# Interrupt Basics

Last updated 6/1/21

# Interrupt Basics

- Interrupts and Exceptions
  - Something has happened and the processor needs to stop what it is currently doing and do something else
  - Interrupts
    - Externally generated
    - Generated by one of the on-chip peripherals
    - Generated by one of the I/O pins
      - Can be dedicated as an interrupt
      - Can be programmed to be an interrupt
  - Exception
    - Internally generated
    - Power failure
    - Overheating
    - Invalid Instruction

# Interrupt Basics

- Why Interrupts
  - I/O operations
    - Peripherals and external devices can let the processor know they are ready vs. the processor constantly checking (polling) each device
      - Pin has changed from 0 to 1
      - A/D conversion complete
      - New SPI message available
  - Routine tasks
    - A task can be setup to occur at regular intervals
      - Updating the time
      - Making sure the processor is not lost in a loop (watchdog)

# Interrupt Basics

- Interrupt Terminology
  - Event
    - The activity that creates an interrupt
  - Flag (interrupt bit)
    - A bit in a register to indicate a specific event has occurred
  - Service(ing)
    - Act of running the code associated with the interrupt
  - Priority
    - Order in which events (interrupts) are serviced
  - Nested
    - A higher priority interrupt can “interrupt” an active, lower priority interrupt
  - Maskable vs. non-maskable
    - Maskable – these interrupts can be selectively enabled or disabled
    - Non-maskable – these cannot be disabled

# Interrupt Basics

- Interrupt Terminology
  - Pending
    - An interrupt that has occurred but has not yet been serviced by the processor
    - Several interrupts can be pending at any given time
      - Various rules to determine the priority
  - Active
    - Currently being serviced
  - ISR
    - Interrupt Service Routine
    - The code run when an interrupt is Active
  - Context switch
    - All the activity associated with changing from one task to another

# Interrupt Basics

- Interrupt Mechanics
  - When a specific condition is true (Event occurs)
    - The processor or peripheral sets a FLAG
  - Two methods to determine if a flag has been set
    - Polling
    - Interrupt Servicing (interrupt controller)

# Interrupt Basics

- Interrupt Mechanics - Polling
  - The code is written to check each interrupt flag at the desired frequency
    - Typically, inside a while loop
  - When a flag is detected,  specific actions are taken
    - These actions are part of the program flow
    - A specific function is called, or a branch is taken
  - When the required actions are complete the program continues
  - The programmer controls the “context switch”

# Interrupt Basics

- Interrupt Mechanics – Interrupt Servicing
  - A special piece of hardware is included in the processor that has connections to all the interrupt flags
    - NVIC – Nested Vectored Interrupt Controller
  - Each type of interrupt has its own reset vector
    - Location in memory that holds the code to run when this type of interrupt occurs (ISR)
  - When the NVIC detects a pending interrupt
    - A series of actions are taken
  - Maskable interrupts have an enable bit
  - There is a global interrupt enable bit in the processor

# Interrupt Basics

- Interrupt Mechanics – Interrupt Servicing
  - When the NVIC detects a pending interrupt
    - The processor completes the current instruction
    - If the pending interrupt is enabled AND
      the new interrupt is the highest priority pending/active interrupt
      - The processor saves the current "state" of the system onto the stack
        - Current value of the PC
        - Critical working registers and status registers
      - The processor loads the associated interrupt (reset) vector into the PC
        - This puts the processor at the beginning of the ISR
        - The processor then operates normally
      - When the ISR is complete
        - The processor returns the saved values from the stack
        - Returning the system back to where it was before the interrupt

Context switch

Context switch

# Interrupt Basics

## Interrupt Process

### Normal execution

**Program Memory**

| Address (hex) | Instruction (hex) |
|---|---|
| 0000 0000 | 0001 1000 |
| 0000 0004 | 0001 1010 |
| 0000 0008 | 0001 1020 |
| 0000 000C | 0001 1040 |
| 0000 0010 | 0001 1060 |
| | |
| … | |
| | |
| 0000 1100 | A23B 1254 |
| 0000 1102 | 23F4 C298 |
| 0000 1104 | D246 7531 |
| 0000 1106 | 5679 0023 |
| | |
| … | |
| | |
| 0001 1018 | 0024 7654 |
| 0001 101C | 9876 5432 |
| 0001 1020 | AABC 4680 |
| 0001 1022 | 9854 2378 |
| 0001 1024 | A765 567A |
| 0001 1026 | 3490 6521 |
| 0001 1028 | BB34 76CC |
| 0001 102A | 8732 2378 |

PC → 0000 1102

active instruction ← 0000 1100

**Data Memory**

| (hex) | (hex) |
|---|---|
| 2000 0000 | 2356 763A |
| 2000 0004 | 23DC 43FA |
| 2000 0008 | 98FC ACF4 |
| | |
| … | |
| | |
| 2000 1000 | 0000 2323 |
| 2000 1004 | 2222 5555 |
| 2000 1008 | 5454 3434 |
| 2000 100C | 0909 0000 |
| | |
| … | |
| | |
| 2000 FFD8 | X |
| 2000 FFD4 | X |
| 2000 FFD8 | X |
| 2000 FFDC | X |
| … | |
| 2000 FFF4 | X |
| 2000 FFF8 | A345 B543 |
| 2000 FFFC | 1245 6789 |

SP → 2000 FFF8

# Interrupt Basics

## Interrupt Process

### Normal execution

**Program Memory**

| Address (hex) | Instruction (hex) |
|---|---|
| 0000 0000 | 0001 1000 |
| 0000 0004 | 0001 1010 |
| 0000 0008 | 0001 1020 |
| 0000 000C | 0001 1040 |
| 0000 0010 | 0001 1060 |
| | |
| … | |
| | |
| 0000 1100 | A23B 1254 |
| 0000 1102 | 23F4 C298 |
| 0000 1104 | D246 7531 |
| 0000 1106 | 5679 0023 |
| | |
| … | |
| | |
| 0001 1018 | 0024 7654 |
| 0001 101C | 9876 5432 |
| 0001 1020 | AABC 4680 |
| 0001 1022 | 9854 2378 |
| 0001 1024 | A765 567A |
| 0001 1026 | 3490 6521 |
| 0001 1028 | BB34 76CC |
| 0001 102A | 8732 2378 |

PC → 0000 1104

active instruction ← 0000 1102

**Data Memory**

| (hex) | (hex) |
|---|---|
| 2000 0000 | 2356 763A |
| 2000 0004 | 23DC 43FA |
| 2000 0008 | 98FC ACF4 |
| | |
| … | |
| | |
| 2000 1000 | 0000 2323 |
| 2000 1004 | 2222 5555 |
| 2000 1008 | 5454 3434 |
| 2000 100C | 0909 0000 |
| | |
| … | |
| | |
| 2000 FFD8 | X |
| 2000 FFD4 | X |
| 2000 FFD8 | X |
| 2000 FFDC | X |
| … | |
| 2000 FFF4 | X |
| 2000 FFF8 | A345 B543 |
| 2000 FFFC | 1245 6789 |

SP → 2000 FFF8

# Interrupt Basics

## Interrupt Process

**Program Memory**

| Address (hex) | Instruction (hex) |
|---|---|
| 0000 0000 | 0001 1000 |
| 0000 0004 | 0001 1010 |
| 0000 0008 | 0001 1020 |
| 0000 000C | 0001 1040 |
| 0000 0010 | 0001 1060 |
| | |
| … | |
| | |
| 0000 1100 | A23B 1254 |
| 0000 1102 | 23F4 C298 |
| 0000 1104 | D246 7531 |
| 0000 1106 | 5679 0023 |
| | |
| … | |
| | |
| 0001 1018 | 0024 7654 |
| 0001 101C | 9876 5432 |
| 0001 1020 | AABC 4680 |
| 0001 1022 | 9854 2378 |
| 0001 1024 | A765 567A |
| 0001 1026 | 3490 6521 |
| 0001 1028 | BB34 76CC |
| 0001 102A | 8732 2378 |

PC → 0000 1104

active instruction ← 0000 1102

### Beginning of Interrupt

Flag set

Interrupt → pending

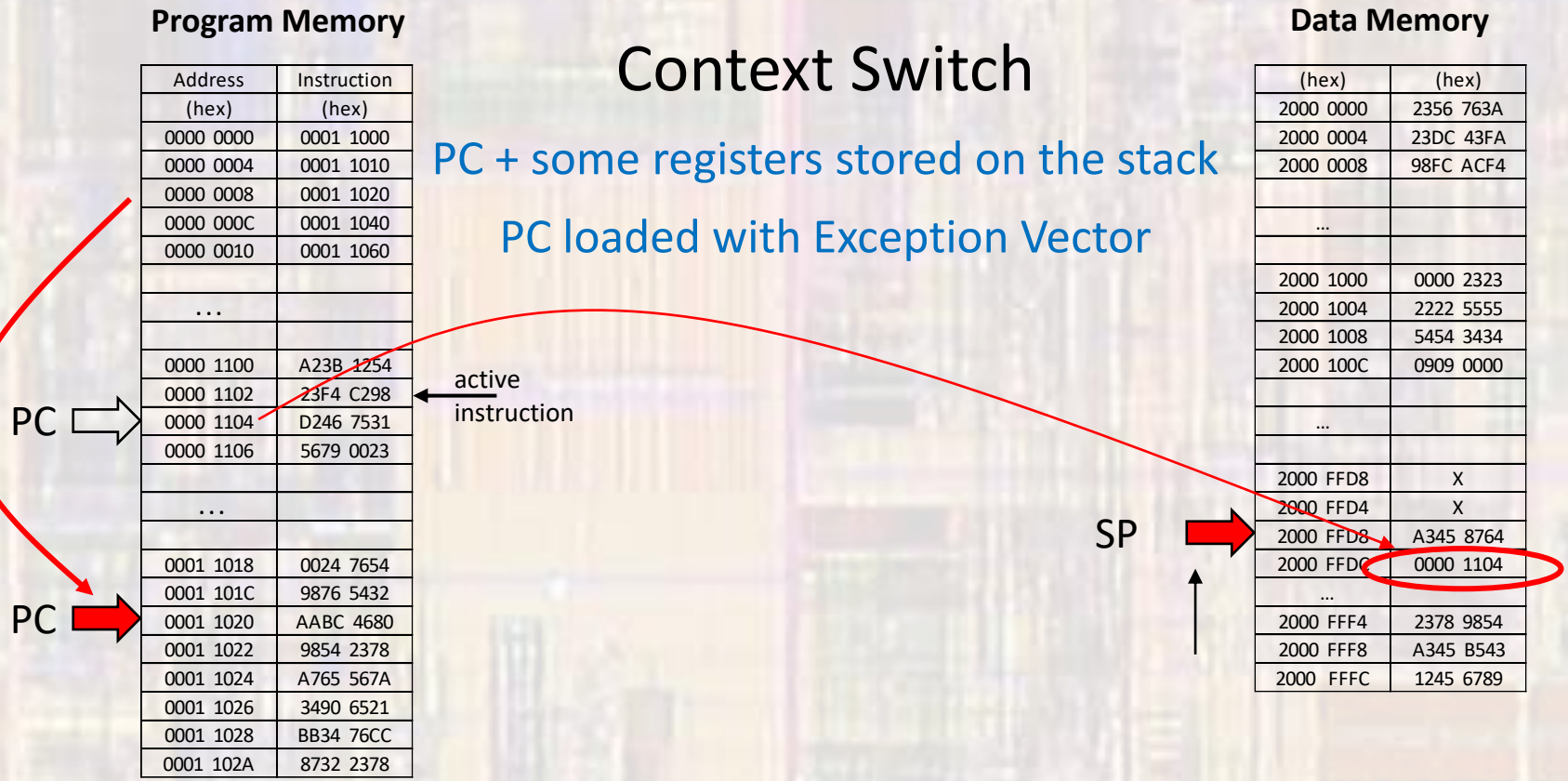
**Data Memory**

| (hex) | (hex) |
|---|---|
| 2000 0000 | 2356 763A |
| 2000 0004 | 23DC 43FA |
| 2000 0008 | 98FC ACF4 |
| | |
| … | |
| | |
| 2000 1000 | 0000 2323 |
| 2000 1004 | 2222 5555 |
| 2000 1008 | 5454 3434 |
| 2000 100C | 0909 0000 |
| | |
| … | |
| | |
| 2000 FFD8 | X |
| 2000 FFD4 | X |
| 2000 FFD8 | X |
| 2000 FFDC | X |
| … | |
| 2000 FFF4 | X |
| 2000 FFF8 | A345 B543 |
| 2000 FFFC | 1245 6789 |

SP → 2000 FFF8

# Interrupt Basics

## Interrupt Process



# Interrupt Basics

## Interrupt Process

**Program Memory**

| Address (hex) | Instruction (hex) |
|---|---|
| 0000 0000 | 0001 1000 |
| 0000 0004 | 0001 1010 |
| 0000 0008 | 0001 1020 |
| 0000 000C | 0001 1040 |
| 0000 0010 | 0001 1060 |
| | |
| … | |
| | |
| 0000 1100 | A23B 1254 |
| 0000 1102 | 23F4 C298 |
| 0000 1104 | D246 7531 |
| 0000 1106 | 5679 0023 |
| | |
| … | |
| | |
| 0001 1018 | 0024 7654 |
| 0001 101C | 9876 5432 |
| 0001 1020 | AABC 4680 |
| 0001 1022 | 9854 2378 |
| 0001 1024 | A765 567A |
| 0001 1026 | 3490 6521 |
| 0001 1028 | BB34 76CC |
| 0001 102A | 8732 2378 |

PC

active instruction

Interrupt Service Routine (ISR)

**Data Memory**

| (hex) | (hex) |
|---|---|
| 2000 0000 | 2356 763A |
| 2000 0004 | 23DC 43FA |
| 2000 0008 | 98FC ACF4 |
| | |
| … | |
| | |
| 2000 1000 | 0000 2323 |
| 2000 1004 | 2222 5555 |
| 2000 1008 | 5454 3434 |
| 2000 100C | 0909 0000 |
| | |
| … | |
| | |
| 2000 FFD8 | X |
| 2000 FFD4 | X |
| 2000 FFD8 | A345 8764 |
| 2000 FFDC | 0000 1104 |
| … | |
| 2000 FFF4 | 2378 9854 |
| 2000 FFF8 | A345 B543 |
| 2000 FFFC | 1245 6789 |

SP

# Interrupt Basics

## Interrupt Process

**Program Memory**

| Address (hex) | Instruction (hex) |
|---|---|
| 0000 0000 | 0001 1000 |
| 0000 0004 | 0001 1010 |
| 0000 0008 | 0001 1020 |
| 0000 000C | 0001 1040 |
| 0000 0010 | 0001 1060 |
| | |
| … | |
| | |
| 0000 1100 | A23B 1254 |
| 0000 1102 | 23F4 C298 |
| 0000 1104 | D246 7531 |
| 0000 1106 | 5679 0023 |
| | |
| … | |
| | |
| 0001 1018 | 0024 7654 |
| 0001 101C | 9876 5432 |
| 0001 1020 | AABC 4680 |
| 0001 1022 | 9854 2378 |
| 0001 1024 | A765 567A |
| 0001 1026 | 3490 6521 |
| 0001 1028 | BB34 76CC |
| 0001 102A | 8732 2378 |

PC

active instruction

Interrupt Service Routine (ISR)

**Data Memory**

| (hex) | (hex) |
|---|---|
| 2000 0000 | 2356 763A |
| 2000 0004 | 23DC 43FA |
| 2000 0008 | 98FC ACF4 |
| | |
| … | |
| | |
| 2000 1000 | 0000 2323 |
| 2000 1004 | 2222 5555 |
| 2000 1008 | 5454 3434 |
| 2000 100C | 0909 0000 |
| | |
| … | |
| | |
| 2000 FFD8 | X |
| 2000 FFD4 | X |
| 2000 FFD8 | A345 8764 |
| 2000 FFDC | 0000 1104 |
| … | |
| 2000 FFF4 | 2378 9854 |
| 2000 FFF8 | A345 B543 |
| 2000 FFFC | 1245 6789 |

SP

# Interrupt Basics

## Interrupt Process

### Return from Interrupt

**Program Memory**

| Address (hex) | Instruction (hex) |
|---|---|
| 0000 0000 | 0001 1000 |
| 0000 0004 | 0001 1010 |
| 0000 0008 | 0001 1020 |
| 0000 000C | 0001 1040 |
| 0000 0010 | 0001 1060 |
| | |
| … | |
| | |
| 0000 1100 | A23B 1254 |
| 0000 1102 | 23F4 C298 |
| 0000 1104 | D246 7531 |
| 0000 1106 | 5679 0023 |
| | |
| … | |
| | |
| 0001 1018 | 0024 7654 |
| 0001 101C | 9876 5432 |
| 0001 1020 | AABC 4680 |
| 0001 1022 | 9854 2378 |
| 0001 1024 | A765 567A |
| 0001 1026 | 3490 6521 |
| 0001 1028 | BB34 76CC |
| 0001 102A | 8732 2378 |

PC → 0001 1024

active instruction → 0001 1022

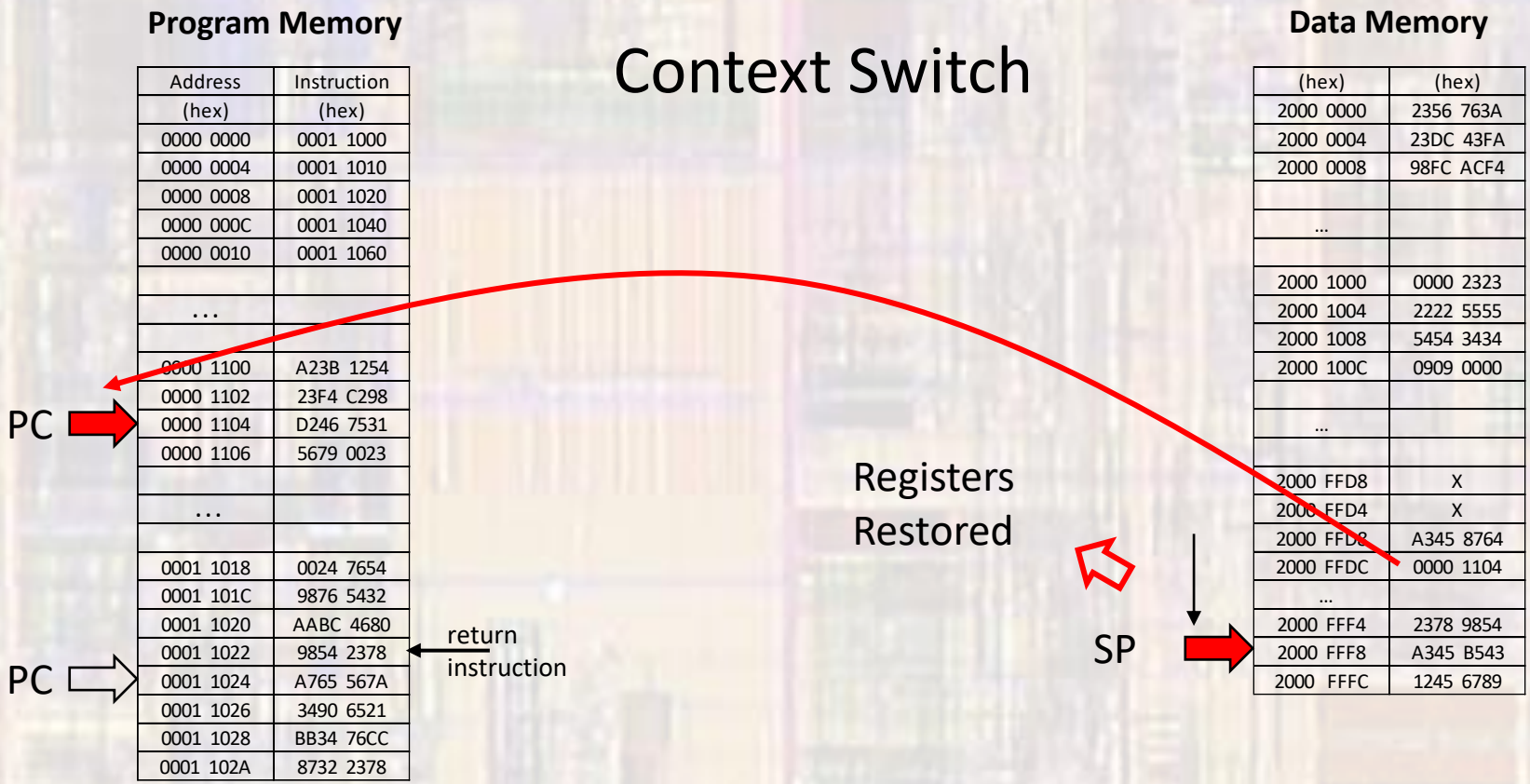
**Data Memory**

| (hex) | (hex) |
|---|---|
| 2000 0000 | 2356 763A |
| 2000 0004 | 23DC 43FA |
| 2000 0008 | 98FC ACF4 |
| | |
| … | |
| | |
| 2000 1000 | 0000 2323 |
| 2000 1004 | 2222 5555 |
| 2000 1008 | 5454 3434 |
| 2000 100C | 0909 0000 |
| | |
| … | |
| | |
| 2000 FFD8 | X |
| 2000 FFD4 | X |
| 2000 FFD8 | A345 8764 |
| 2000 FFDC | 0000 1104 |
| … | |
| 2000 FFF4 | 2378 9854 |
| 2000 FFF8 | A345 B543 |
| 2000 FFFC | 1245 6789 |

SP → 2000 FFD8

# Interrupt Basics

## Interrupt Process



# Interrupt Basics

## Interrupt Process

### Normal Execution

**Program Memory**

| Address (hex) | Instruction (hex) |
|---|---|
| 0000 0000 | 0001 1000 |
| 0000 0004 | 0001 1010 |
| 0000 0008 | 0001 1020 |
| 0000 000C | 0001 1040 |
| 0000 0010 | 0001 1060 |
| | |
| … | |
| | |
| 0000 1100 | A23B 1254 |
| 0000 1102 | 23F4 C298 |
| 0000 1104 | D246 7531 |
| 0000 1106 | 5679 0023 |
| | |
| … | |
| | |
| 0001 1018 | 0024 7654 |
| 0001 101C | 9876 5432 |
| 0001 1020 | AABC 4680 |
| 0001 1022 | 9854 2378 |
| 0001 1024 | A765 567A |
| 0001 1026 | 3490 6521 |
| 0001 1028 | BB34 76CC |
| 0001 102A | 8732 2378 |

PC → 0000 1106

active instruction → 0000 1104

**Data Memory**

| (hex) | (hex) |
|---|---|
| 2000 0000 | 2356 763A |
| 2000 0004 | 23DC 43FA |
| 2000 0008 | 98FC ACF4 |
| | |
| … | |
| | |
| 2000 1000 | 0000 2323 |
| 2000 1004 | 2222 5555 |
| 2000 1008 | 5454 3434 |
| 2000 100C | 0909 0000 |
| | |
| … | |
| | |
| 2000 FFD8 | X |
| 2000 FFD4 | X |
| 2000 FFD8 | A345 8764 |
| 2000 FFDC | 0000 1104 |
| … | |
| 2000 FFF4 | 2378 9854 |
| 2000 FFF8 | A345 B543 |
| 2000 FFFC | 1245 6789 |

SP → 2000 FFF8